Stereopsis

Raul Queiroz Feitosa

Objetive

This chapter introduces the basic techniques for a 3 dimensional scene reconstruction based on a set of projections of individual points on two calibrated cameras.

Content

- Introduction
- Reconstruction
- The Correspondence Problem
- Rectification

Introduction

Correspondence

- Detection of corresponding points in a pair of stereo images.

Reconstruction

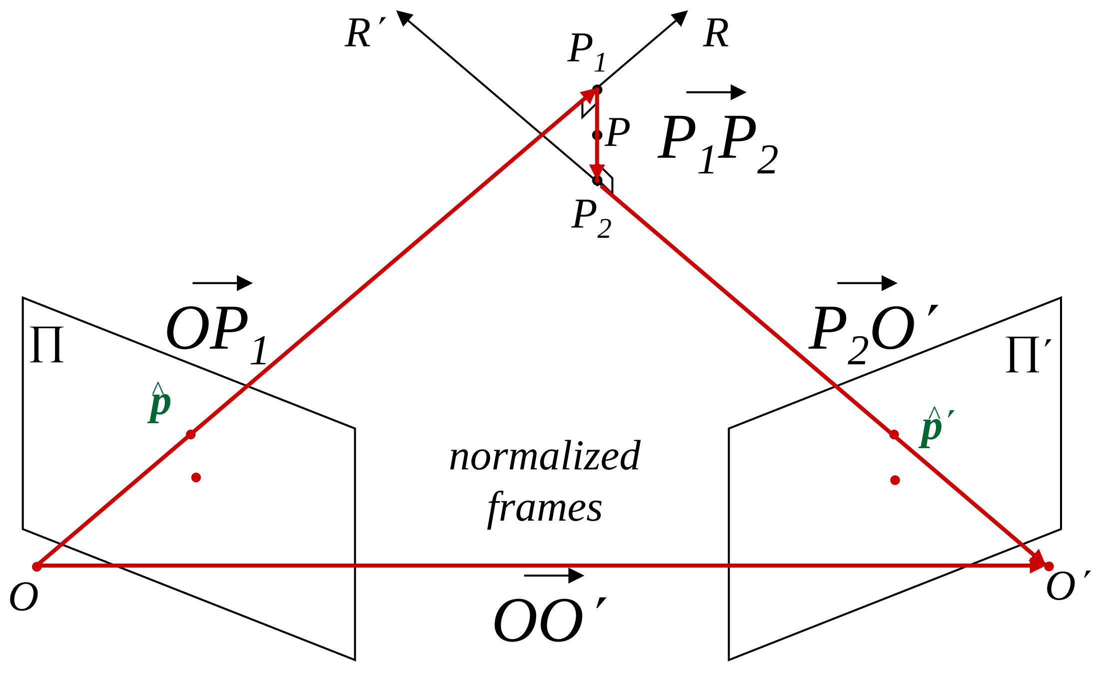
- Based on a set of corresponding points, compute its 3D position in the world.

Content

- Introduction
- Reconstruction
- The Correspondence Problem
- Rectification

Midpoint Triangulation

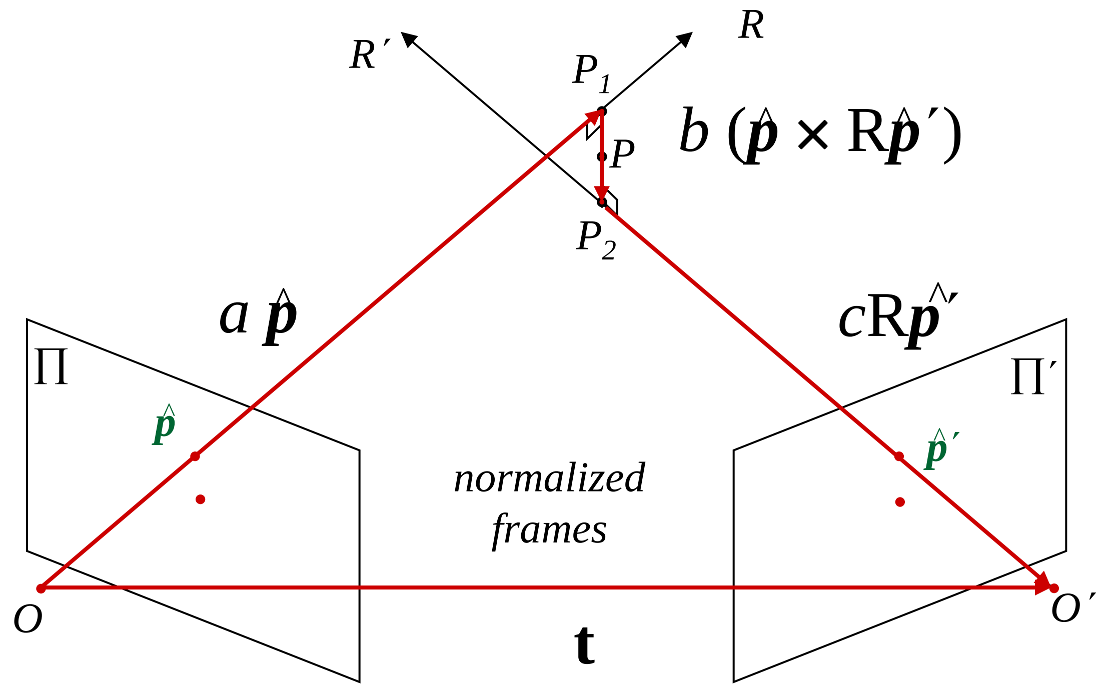
$$\overrightarrow{OP_1} + \overrightarrow{P_1P_2} + \overrightarrow{P_2O'} = \overrightarrow{OO'}$$



Midpoint Triangulation

Expressing the equation in the normalized left camera frame, yields:

$$a\,\hat{\boldsymbol{p}} + b\,(\hat{\boldsymbol{p}} \times \mathrm{R}\hat{\boldsymbol{p}}') + c\,\mathrm{R}\,\hat{\boldsymbol{p}}' = \mathbf{t} \qquad (a, b \text{ and } c \text{ can be computed})$$



where R is the rotation matrix of the right camera frame and **t** is the vector connecting the principal points, both in relation to the left camera frame.

Midpoint Triangulation

Algorithm

Given $\boldsymbol{p}$, $\boldsymbol{p}'$, $\mathcal{K}$, $\mathcal{R}$, $\boldsymbol{t}$, $\mathcal{K}'$, $\mathcal{R}'$ and $\boldsymbol{t}'$

1. Compute

$$\hat{\boldsymbol{p}} = \mathcal{K}^{-1}\boldsymbol{p}$$

$$\hat{\boldsymbol{p}}' = \mathcal{K}'^{-1}\boldsymbol{p}'$$

$\mathrm{R} = \mathcal{R}\mathcal{R}'^{-1}$ ($\rightarrow$ rotation relative to the left camera frame)

$\mathbf{t} = -\mathcal{R}\mathcal{R}'^{-1}\boldsymbol{t}' + \boldsymbol{t}$ ($\rightarrow$ translation relative to the left camera frame)

Midpoint Triangulation

Algorithm (cont.)

2. Compute a, b and c such that

$$a\,\hat{\boldsymbol{p}} + b\,(\hat{\boldsymbol{p}} \times \mathrm{R}\hat{\boldsymbol{p}}') + c\,\mathrm{R}\hat{\boldsymbol{p}}' = \mathbf{t}$$

3. Compute ${}^{C}P$ in the left camera frame with

$${}^{C}\boldsymbol{P} = a\,\hat{\boldsymbol{p}} + b\,(\hat{\boldsymbol{p}} \times \mathrm{R}\hat{\boldsymbol{p}}')/2$$

4. If you wish, transform ${}^{C}P$ to the world frame,

$${}^{W}\boldsymbol{P} = \mathcal{R}^{-1}\,({}^{C}\boldsymbol{P} - \boldsymbol{t})$$

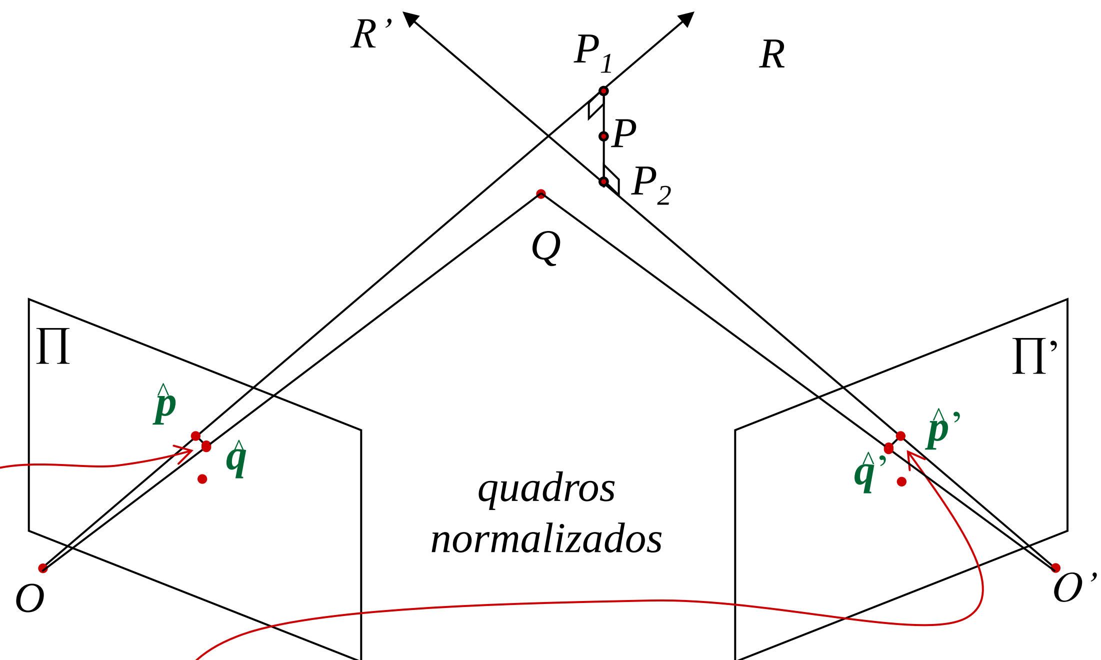
$\boldsymbol{t} \neq \mathbf{t}$

Linear Reconstruction

$$
\begin{aligned}
z\,\boldsymbol{p} &= \mathcal{M}\,\boldsymbol{P} \quad \Rightarrow \quad \boldsymbol{p} \times \mathcal{M}\,\boldsymbol{P} = \boldsymbol{0} \\
z'\,\boldsymbol{p}' &= \mathcal{M}'\,\boldsymbol{P} \quad \Rightarrow \quad \boldsymbol{p}' \times \mathcal{M}'\,\boldsymbol{P} = \boldsymbol{0}
\end{aligned}
\quad \Leftrightarrow \quad
\begin{pmatrix} [\boldsymbol{p}_{\times}]\,\mathcal{M} \\ [\boldsymbol{p}'_{\times}]\,\mathcal{M}' \end{pmatrix} \boldsymbol{P} = \boldsymbol{0}
$$

- It is a system with 4 independent linear equations ($[\boldsymbol{p}_{\times}]$ e $[\boldsymbol{p}'_{\times}]$ have *rank* 2).
- It can be applied for more than 2 cameras.

Geometric Reconstruction

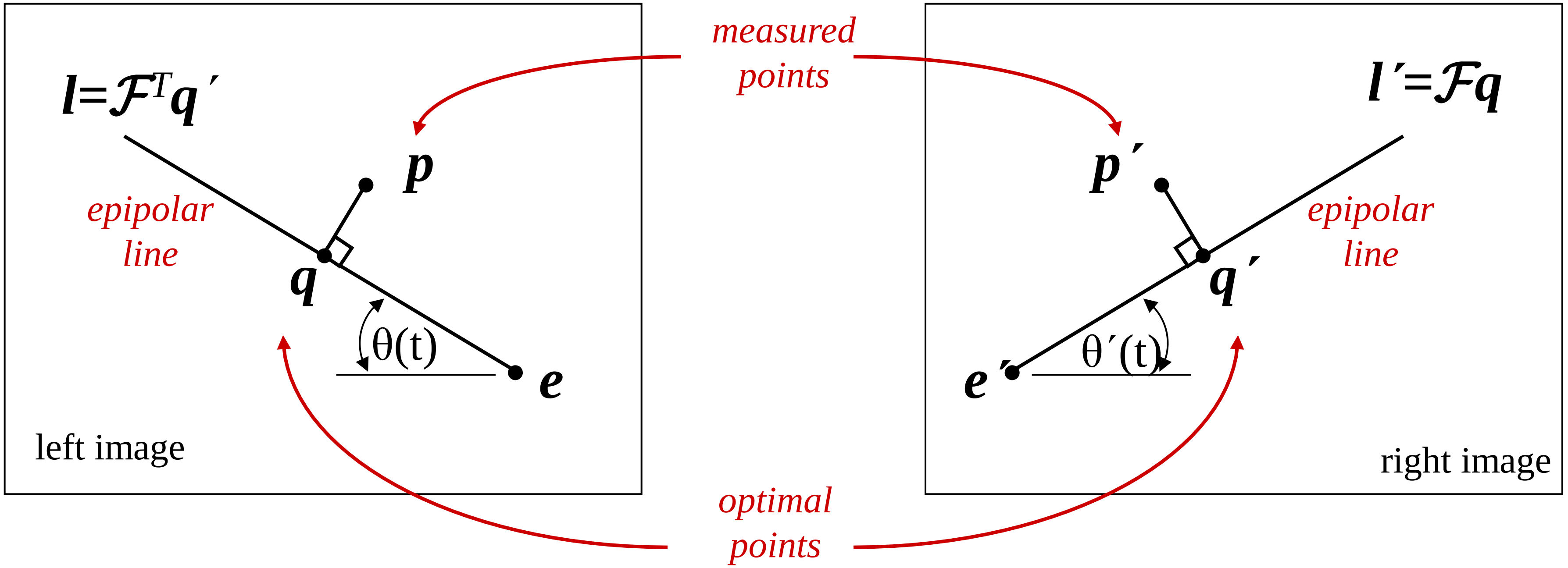


The point Q with images $\boldsymbol{q}$ e $\boldsymbol{q}$' where the rays do intercept are such that $d^2(\mathbf{p},\mathbf{q}) + d^2(\mathbf{p}',\mathbf{q}')$ under $\mathbf{q}^T \mathcal{F}\, \mathbf{q}' = 0$ is minimum → a non linear system.

Geometric Reconstruction

Algorithm – geometric interpretation

The noise free projections lie on a pair of corresponding epipolar lines



Geometric Reconstruction

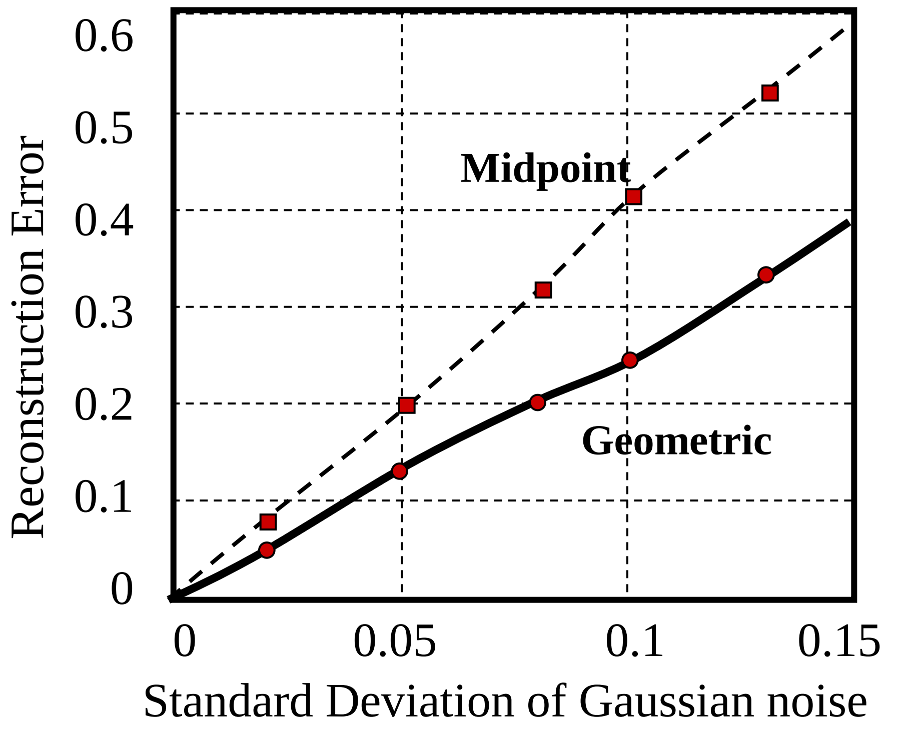
Algorithm outline

1. Parameterize the pencil of epipolar lines in the left image by a parameter $t \rightarrow \boldsymbol{l}(t)$
2. Using the fundamental matrix $\mathcal{F}$, compute the corresponding epipolar line in the right image $\rightarrow \boldsymbol{l}'(t)$
3. Express the distance $d^2(p,\boldsymbol{l}(t)) + d^2(p',\boldsymbol{l}'(t))$ explicitly as function of t $\rightarrow$ a polynomial of degree 6.
4. Find the value of t that minimizes this distance.
5. Compute q and q' and from them Q.

Details can be found in

Multiple View Geometry in computer Vision, Hartley,R. and Zisserman, A., pp 315…

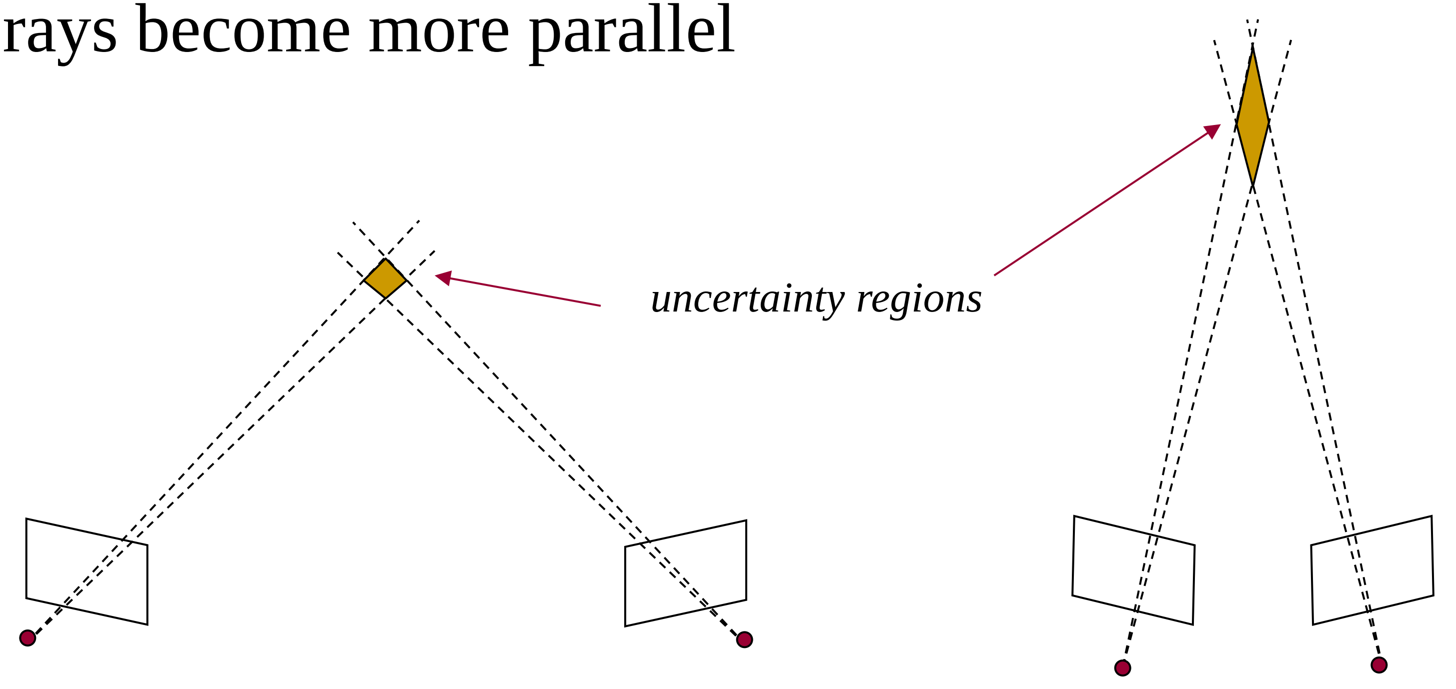
Evaluation on Real Images



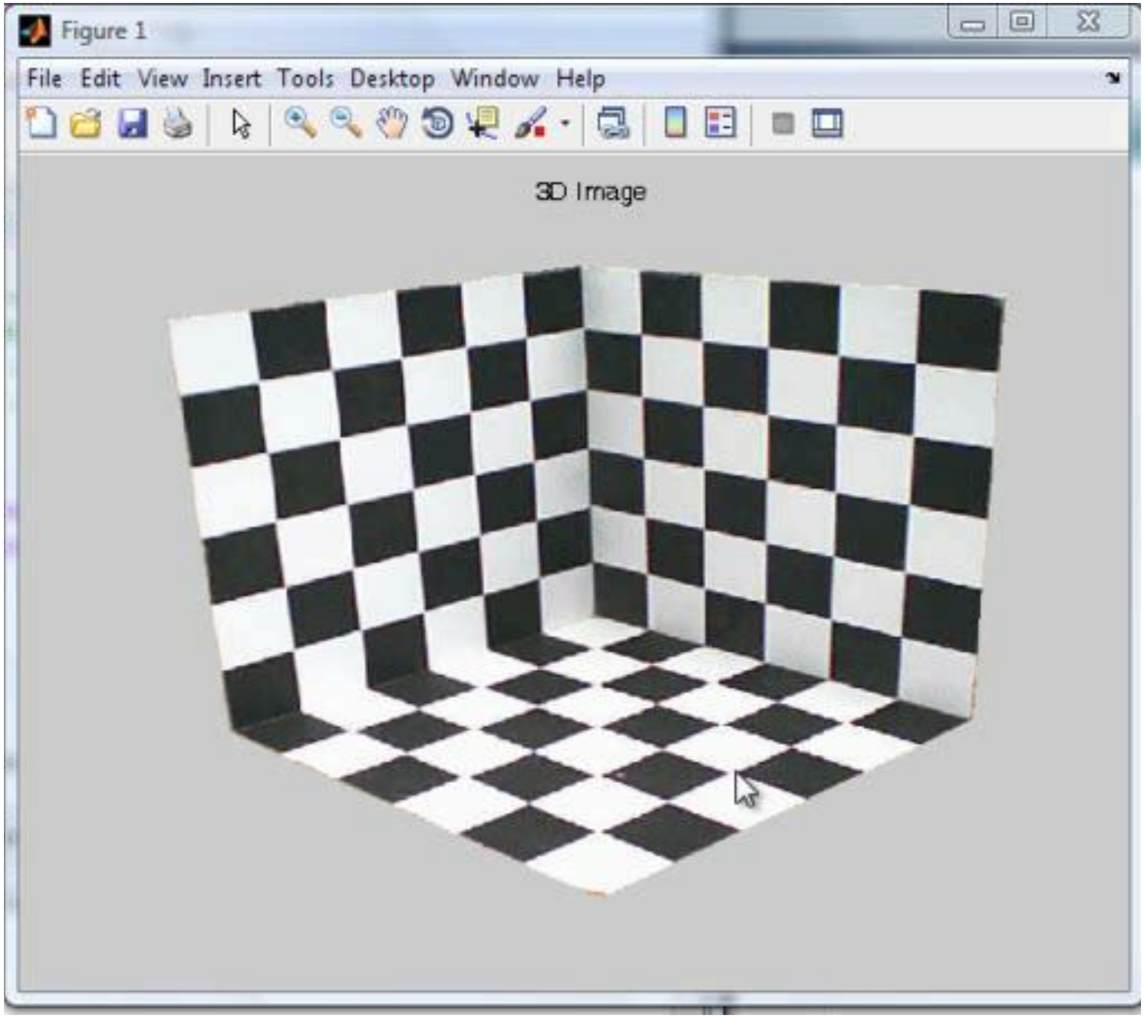
From
Multiple View Geometry in computer Vision, Hartley,R. and Zisserman, A., pp 315…

Evaluation on Real Images

Points are less precisely located along the rays as the rays become more parallel



Texturing 3D Models



[watch]

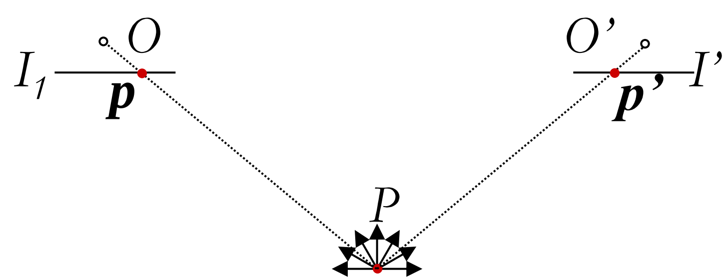
Content

- Introduction
- Reconstruction
- The Correspondence Problem
- Rectification

The Correspondence Problem

Introduction

- Assume the point P in space emits light with the same energy in all directions (i.e. the surface around P is Lambertian)



restrictive assumption

then

$$I(\boldsymbol{p}) = I'(\boldsymbol{p}') \qquad \textit{(brightness constancy constraint)}$$

where $\boldsymbol{p}$ and $\boldsymbol{p}'$ are the images of P in the views I and I'.

- The correspondence problem consists of establishing relationships between $\boldsymbol{p}$ and $\boldsymbol{p}'$, i.e.

$$I(\boldsymbol{p}) = I'(\, h(\boldsymbol{p})\,) \qquad (\, h \rightarrow \textit{deformation})$$

The Correspondence Problem

Introduction

- It is difficult to locate an accurate match for points lying in uniform neighborhoods.
- Also difficult is the location of points close to edges.
- Therefore, before we start searching for pairs of corresponding points, it is convenient to establish which points in one image have good characteristics for an accurate match.
- They must be in neighborhoods with high *dynamics*.

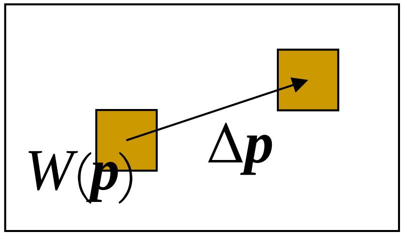
The Correspondence Problem

Local deformation models

We look for transformations h that model the deformation on a domain $W(\boldsymbol{p})$ around $\boldsymbol{p}=[u\ v]^T$.

- Translational model

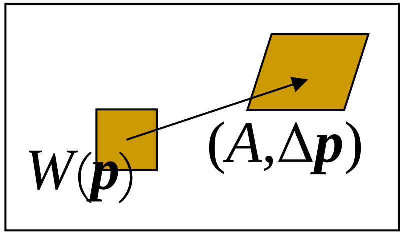
$$h(\boldsymbol{p}) = \boldsymbol{p} + \Delta\boldsymbol{p}$$



- Affine model

$$h(\boldsymbol{p}) = A\,\boldsymbol{p} + \Delta\boldsymbol{p},$$

where $A \in R^{2\times 2}$



- Transformation in intensity values

$$I(\boldsymbol{p}) = I'(\,h(\boldsymbol{p})\,) + \eta(\,h(\boldsymbol{p})\,)$$

The Correspondence Problem

Matching by Sum of Squared Differences (SSD)

- We choose a class of transformation and look for the particular transformation *h* that minimizes the effects of noise integrated over a window $W(\boldsymbol{p})$, i. e.,

$$\boxed{\hat{h} = \arg\min_{h} \sum_{\tilde{\boldsymbol{p}} \in W(\boldsymbol{p})} \left\| I(\tilde{\boldsymbol{p}}) - I'(h(\tilde{\boldsymbol{p}})) \right\|^2}$$

- If $I(\boldsymbol{p})$=constant, for $\boldsymbol{p} \in W(\boldsymbol{p})$, the same happens for I', and the norm being minimized does not depend on *h*!!!! (blank wall effect)

The Correspondence Problem

Matching by Normalized Cross-Correlation (NCC)

- SSD is not invariant to scaling and shifts in image intensities (deviation from the Lambertian assumptions).
- The Normalized Cross-Correlation (NCC)

$$\boxed{NCC(h) = \frac{\sum_{W(\boldsymbol{p})}\left(I(\tilde{\boldsymbol{p}}) - \bar{I}\right)\left(I'(h(\tilde{\boldsymbol{p}})) - \bar{I}'\right)}{\sqrt{\sum_{W(\boldsymbol{p})}\left(I(\tilde{\boldsymbol{p}}) - \bar{I}\right)^2 \sum_{W(\boldsymbol{p})}\left(I'(h(\tilde{\boldsymbol{p}})) - \bar{I}'\right)^2}}}$$

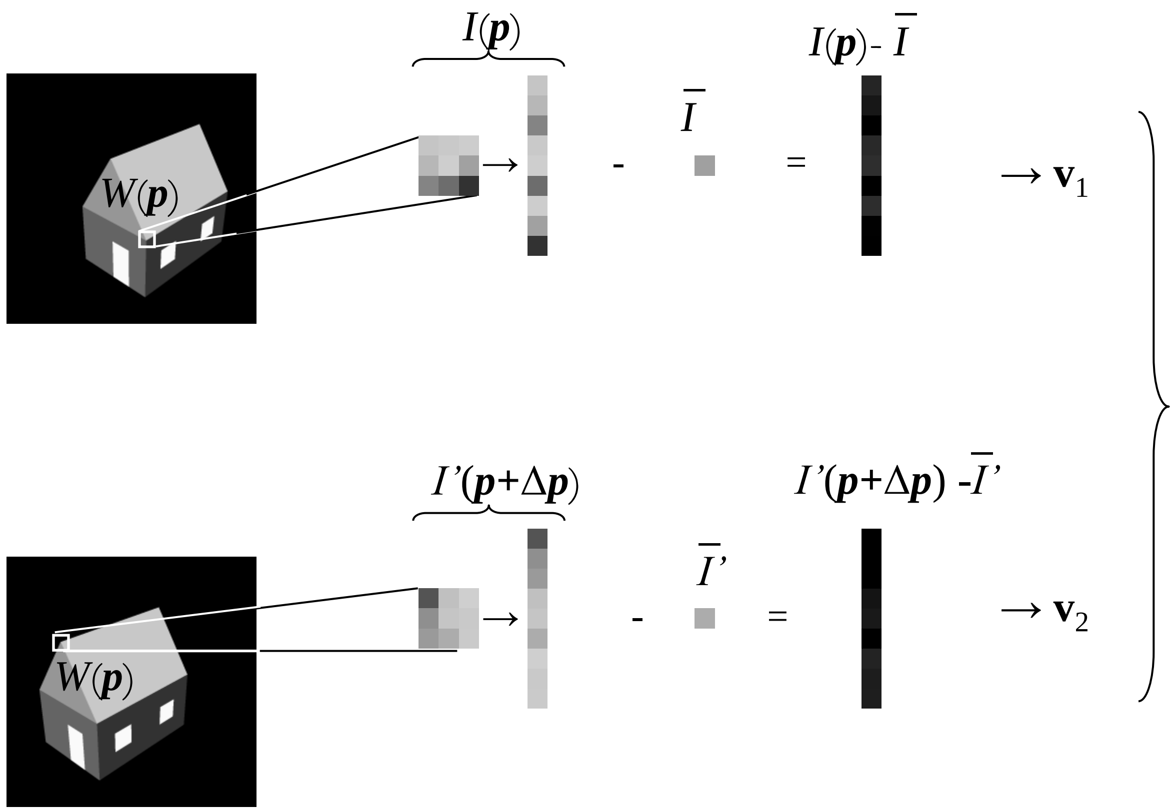
where $\bar{I}$ and $\bar{I}'$ are the mean intensities of *I* and *I'* on $W(\boldsymbol{p})$.

- $-1 \le NCC \le 1$. If =1, $I(\boldsymbol{p})$ and $I'(h(\boldsymbol{p}))$ match perfectly.

The Correspondence Problem

Matching by NCC

- For the translational model $\longrightarrow$ $h(\boldsymbol{p}) = \boldsymbol{p} + \Delta\boldsymbol{p}$.

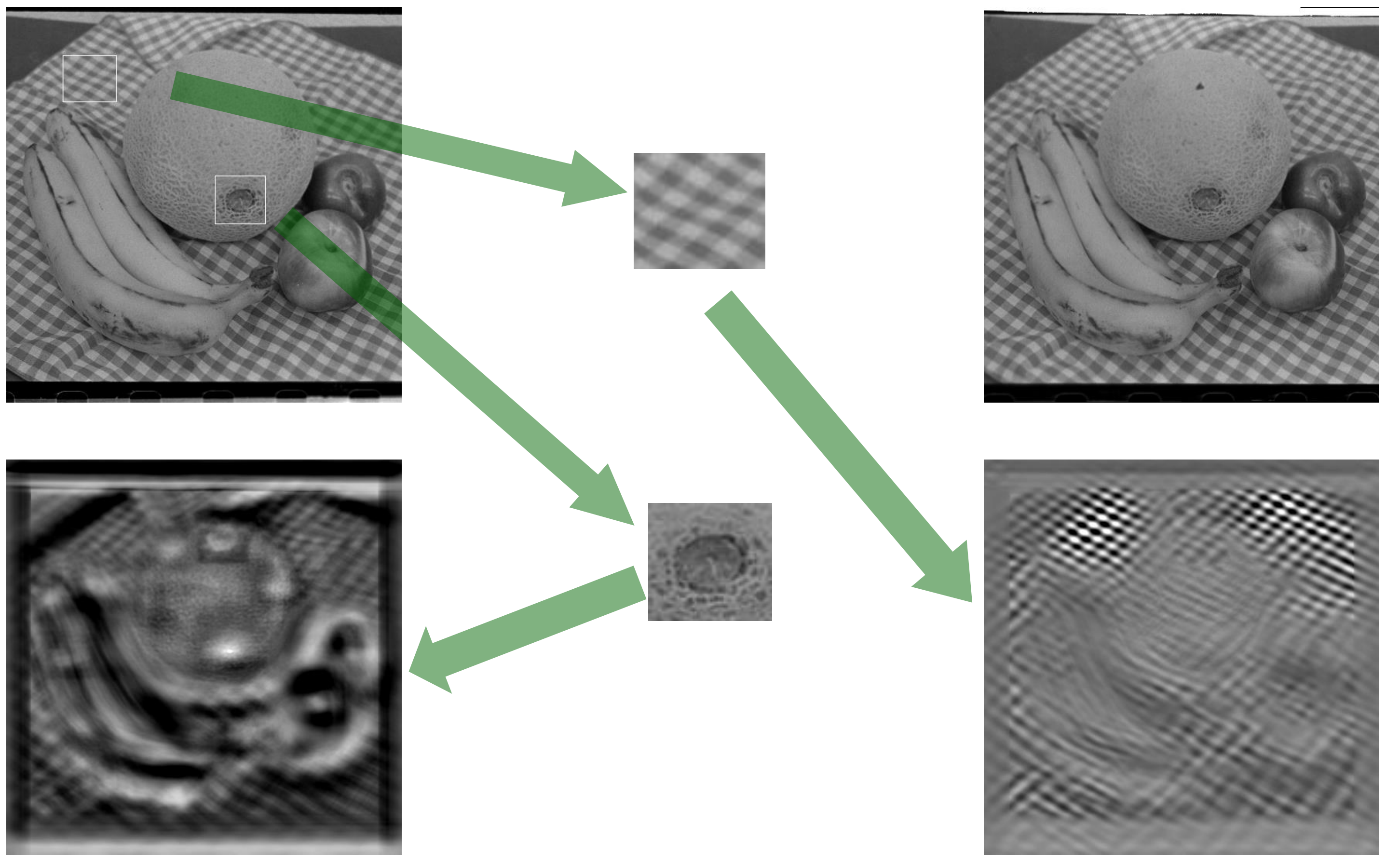


$$NCC = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \, \|\mathbf{v}_2\|}$$

NCC is equal to the cosine between $\mathbf{v}_1$ and $\mathbf{v}_2$

The Correspondence Problem

Matching by NCC (example)



Least Square Matching

Affine model $\longrightarrow$ $\boldsymbol{p}' = h(\boldsymbol{p}) = \mathbf{A}\,\boldsymbol{p} + \Delta\boldsymbol{p}$

Let's assume that

$$\boldsymbol{p}' = \begin{bmatrix} u' \\ v' \end{bmatrix}, \quad \boldsymbol{p} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \Delta\boldsymbol{p} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Thus

$$u' = a_{11}u + a_{12}v + b_1$$

and

$$v' = a_{21}u + a_{22}v + b_2$$

Least Square Matching

Affine model $\longrightarrow$ $\boldsymbol{p}' = h(\boldsymbol{p}) = A\,\boldsymbol{p} + \Delta\boldsymbol{p}$

- Matching is established if (*SSD)*

$$I(u,v) = I'(u',v') \approx I'(u'_0, v'_0) + \frac{\partial I'}{\partial u'}du' + \frac{\partial I'}{\partial v'}dv' \quad \rightarrow$$

$$I(u,v) - I'(u'_0, v'_0) = g_{u'}\left(u'_0 da_{11} + v'_0 da_{12} + db_1\right) + g_{v'}\left(u'_0 da_{21} + v'_0 da_{22} + db_2\right)$$

that can be written in more compact form as …

Least Square Matching

Affine model $\longrightarrow$ $\boldsymbol{p}' = h(\boldsymbol{p}) = A\,\boldsymbol{p} + \Delta\boldsymbol{p}$

- Matching is established if (*SSD)*

$$I(u,v) - I'(u'_0, v'_0) = \mathcal{U}\,\mathbf{x}$$

where

$$\mathcal{U} = \begin{pmatrix} g_{u'}u'_0 & g_{u'}v'_0 & g_{u'} & g_{v'}u'_0 & g_{v'}v'_0 & g_{v'} \end{pmatrix}$$

$$\mathbf{x}^T = \begin{pmatrix} da_{11} & da_{12} & db_1 & da_{21} & da_{22} & db_2 \end{pmatrix}$$

Least Square Matching

Affine model $\longrightarrow$ $\boldsymbol{p}' = h(\boldsymbol{p}) = A\,\boldsymbol{p} + \Delta\boldsymbol{p}$

- Stacking the equation generated by each point in $W(\boldsymbol{p})$ yields

$$\begin{array}{cccc} \mathbf{y} & & \mathcal{W} & \\ \begin{bmatrix} I(u_1, v_1) - I'(u'_{01}, v'_{01}) \\ \vdots \\ I(u_n, v_n) - I'(u'_{0n}, v'_{0n}) \end{bmatrix} & = & \begin{bmatrix} g_{u'} u'_{01} & g_{u'} v'_{01} & g_{u'} & g_{v'} u'_{01} & g_{v'} v'_{01} & g_{v'} \\ & & & \vdots & & \\ g_{u'} u'_{0n} & g_{u'} v'_{0n} & g_{u'} & g_{v'} u'_{0n} & g_{v'} v'_{0n} & g_{v'} \end{bmatrix} & \mathbf{x} \end{array}$$

- It is a over determined system whose solution is given by

$$\mathbf{x} = \mathcal{W}^{\dagger}\,\mathbf{y}$$

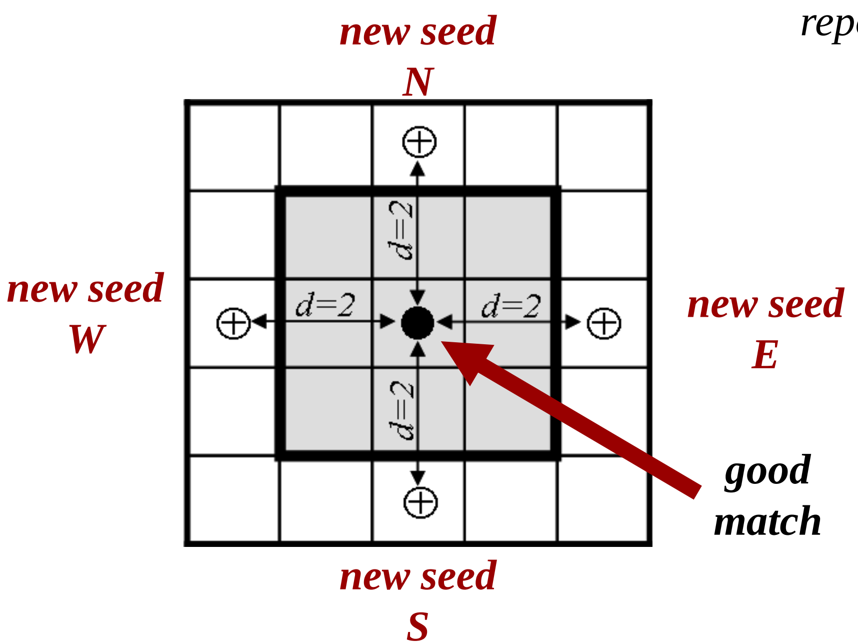
Least Square Matching

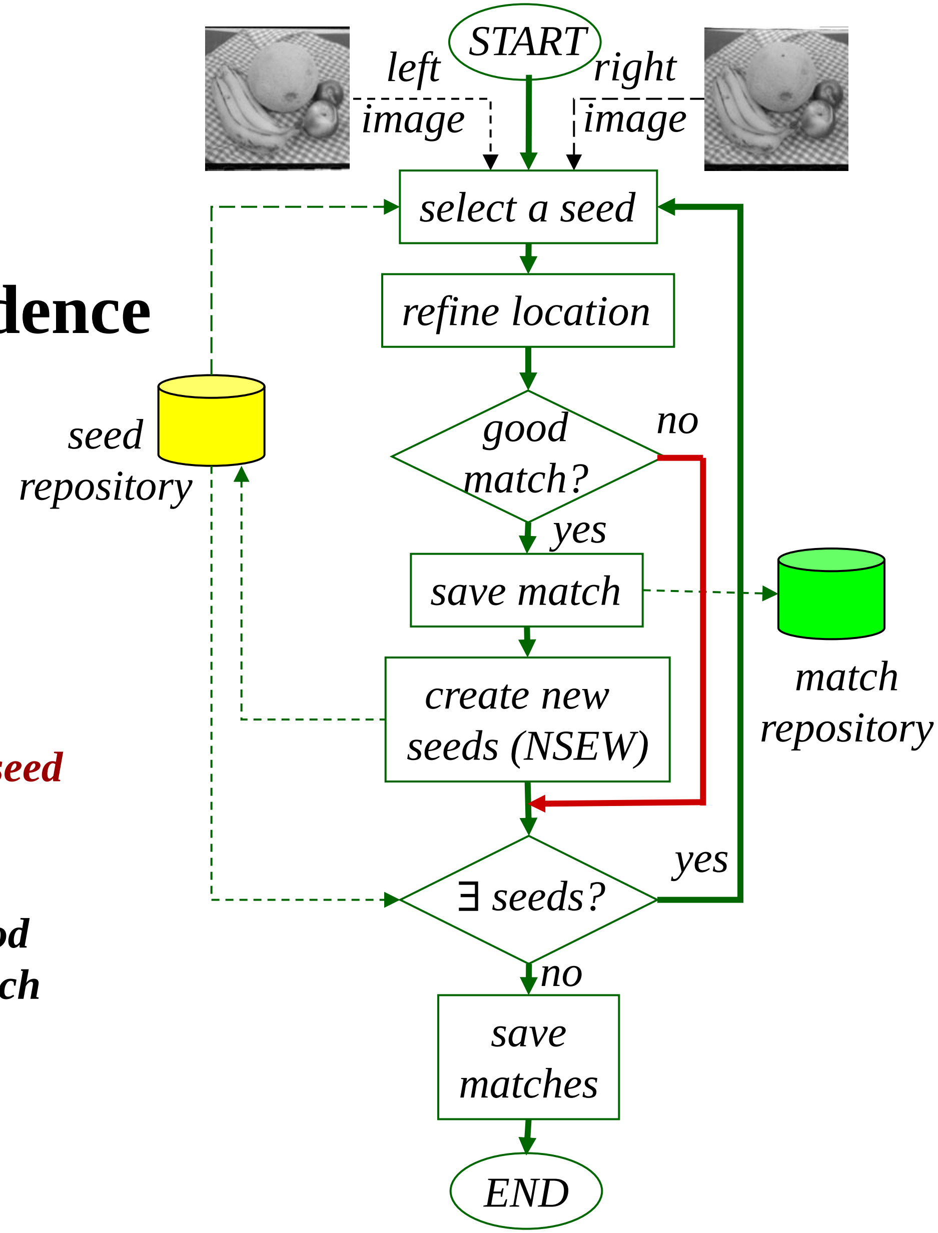
Affine model $\rightarrow$ $\boldsymbol{p}' = h(\boldsymbol{p}) = \mathbf{A}\,\boldsymbol{p} + \Delta\boldsymbol{p}$

1. Start with, k=0 and
$$\mathbf{A}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Delta\boldsymbol{p}^0 = \begin{bmatrix} b_1^0 \\ b_2^0 \end{bmatrix}$$
2. $k=k+1;$ for each (u,v) in W($\boldsymbol{p}$) apply the transformation to compute $I'(u',v')$, $g_{u'}$ and $g_{v'}$.
3. Build the equation (previous slide) and compute **x**
4. Update the parameters
$$\mathbf{A}^{k+1} = \mathbf{A}^k + \begin{bmatrix} da_{11}^k & da_{12}^k \\ da_{21}^k & da_{22}^k \end{bmatrix}, \quad \Delta\boldsymbol{p}^{k+1} = \Delta\boldsymbol{p}^k + \begin{bmatrix} db_1^k \\ db_2^k \end{bmatrix}$$
5. Repeat steps 2 to 4 till ||**x**|| goes below a certain limit.

Region Growing

Building a dense correspondence





Region Growing

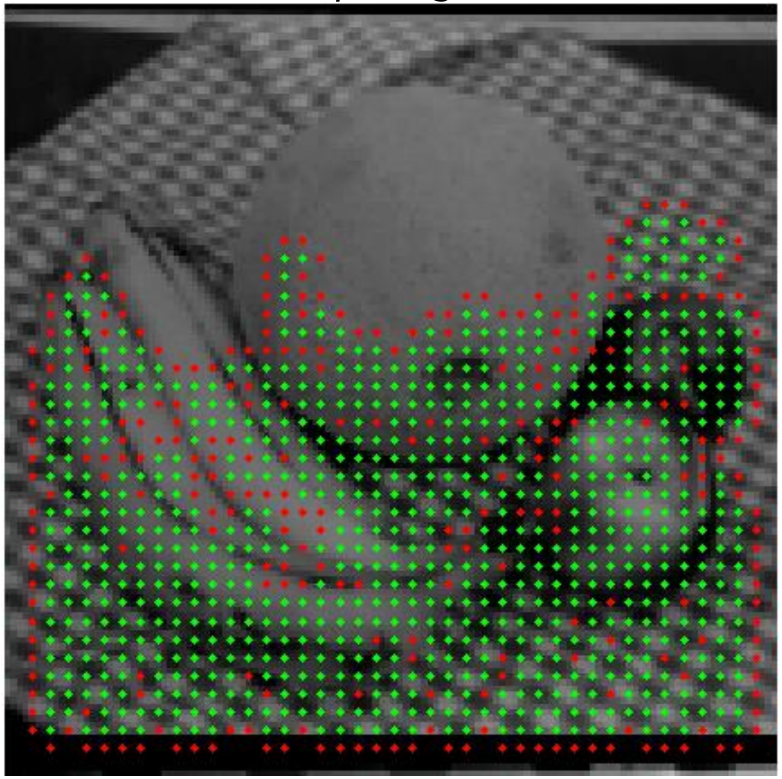
Building a dense correspondence map with Region Growing

1. Select a seed of corresponding points manually, and add it in ***To_Check_List*** .
2. Take the first pair, delete it in ***To_Check_List*** and add it to ***Checked_List*** .
3. Refine location in right image using a correlation based method
4. If constrast > ***min_contrast*** AND ***correlation > min_correlation***,
   - Put pair in the ***Match_List.***
   - Add to the ***To_Check_List*** its neighbors *d* pixels to the North, to the South, to the East and to the West, as long as they belong neither to ***Checked_List*** nor to ***To_Check_List*** .
5. Do steps 2 to 4 till ***To_Check_List*** is empty.

Region Growing

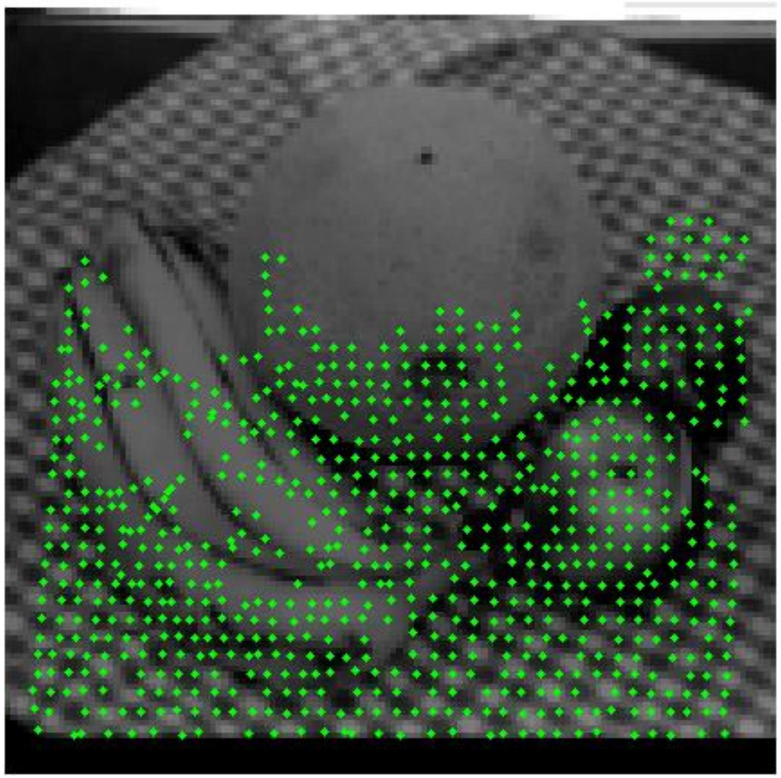
Example

left image



[watch]

right image



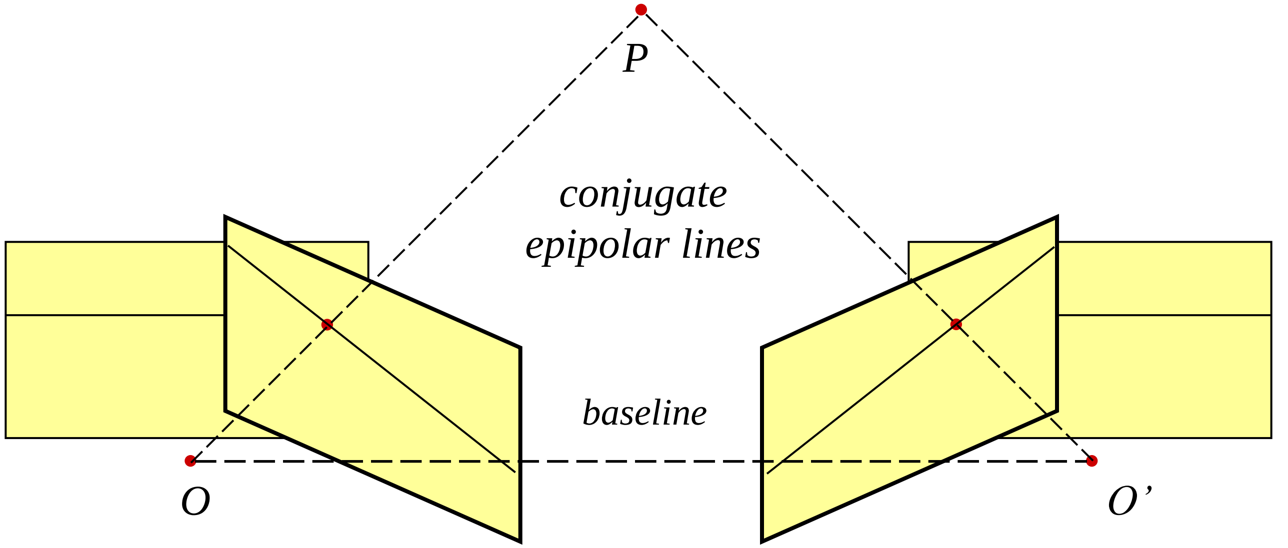
Content

- Introduction
- Reconstruction
- The Correspondence Problem
- Rectification

Rectification

Transforms a pair of stereo images in such a way that

- both images stay on a single plane parallel to the baseline, and
- conjugate epipolar lines are collinear and horizontal.



- The search for corresponding points becomes 1D.

Rectification

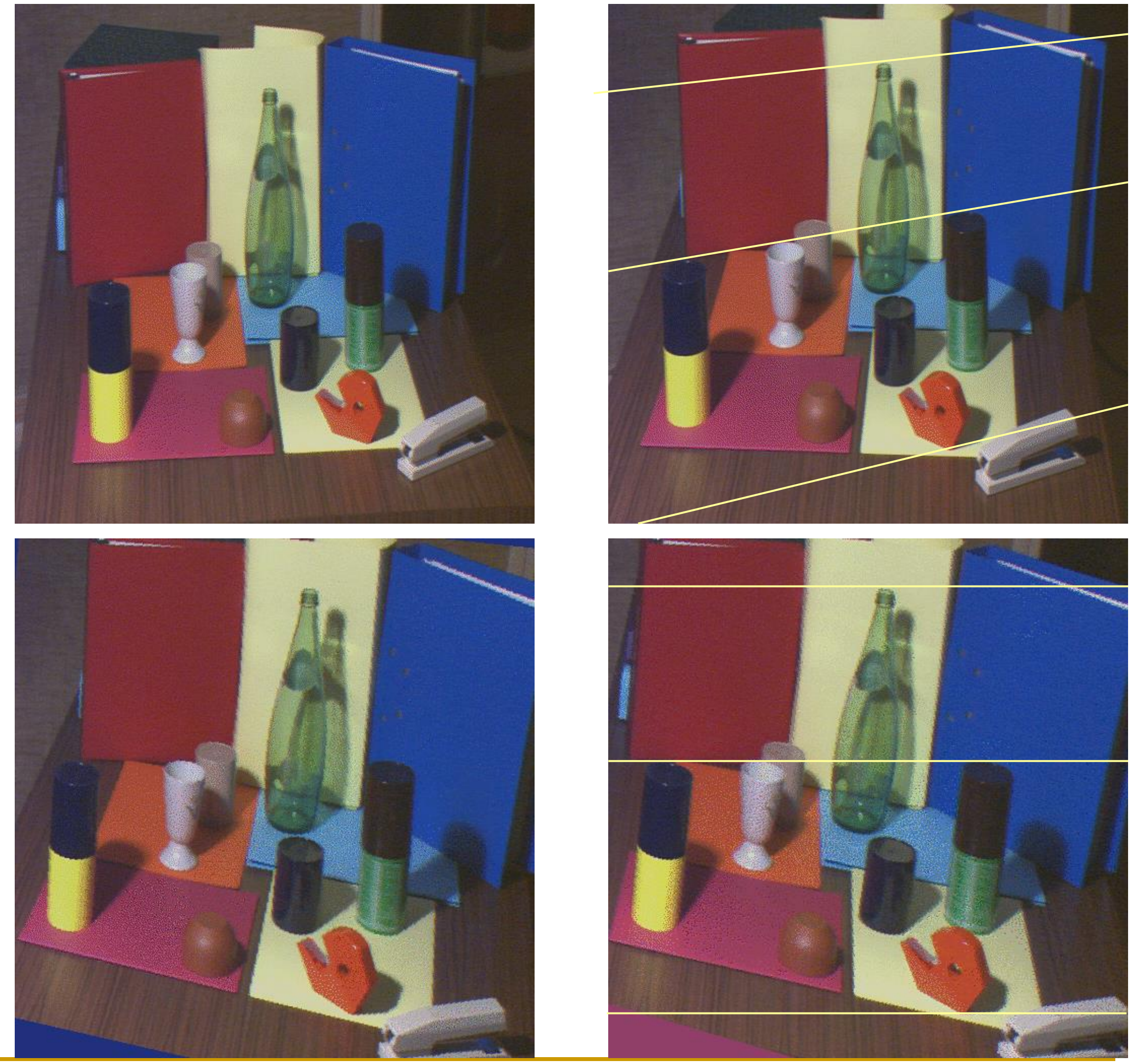
Example:

original

images

rectified

images



Rectification

Find two projective transformations H and H'

$$\tilde{\boldsymbol{p}} = H\,\boldsymbol{p} \quad \text{and} \quad \tilde{\boldsymbol{p}}' = H'\boldsymbol{p}'$$

that applied to each image make matching points have approximately the same *v*-coordinate

Clearly, since the new image planes are parallel to the base line, the **epipoles are in the infinity**.

Recall Projective Geometry

2 Dimensional

- If a point $\boldsymbol{x}$ lies on line $\mathbf{l}$

$$\boldsymbol{x}^T \mathbf{l} = 0$$

- The intersection $\boldsymbol{x}$ of two lines $\mathbf{l}_1$ and $\mathbf{l}_2$

$$\boldsymbol{x} = \mathbf{l}_1 \times \mathbf{l}_2$$

- The line joining points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$

$$\mathbf{l} = \boldsymbol{x}_1 \times \boldsymbol{x}_2$$

- Note that in $\mathbb{P}^2$ parallel lines intersect at infinity

$$(a,b,c)\times(a,b,c') = (\, b(c'-c), a(c-c'), 0)$$

Recall Projective Geometry

2 Dimensional

	$\mathbb{P}^2$	$\mathbb{R}^2$	
points	$\boldsymbol{x} = (kx_1, kx_2, k)^T = (x_1, x_2, 1)^T$ for $k \neq 0$ $\boldsymbol{x}_\infty = (kx_1, kx_2, 0)^T = (x_1, x_2, 0)^T$ for $k \neq 0$	$\boldsymbol{x} = (x_1, x_2)^T$	points at infinity
lines	$\mathbf{l} = (kl_1, kl_2, k)^T = (l_1, l_2, 1)^T$ for $k \neq 0$ $\mathbf{l}_\infty = (0, 0, k)^T$ for $k \neq 0$	$\mathbf{l} = (a, b, c)^T$ -	lines at infinity

Mapping epipole to infinity

Assume that

- u_0 =0 and v_0 =0 (the origin is in the image center)
- epipole $\mathbf{e} = [\,f, 0, 1]^T$ lies on the u-axis.

So, the transformation

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix}$$

takes the epipole to the point at infinity $[\,f, 0, 0]^T$

Mapping epipole to infinit

For an arbitrarily placed point of interest $\boldsymbol{p}_0$ and epipole **e**, the mapping H that makes epipolar lines run parallel to the u axis is $H=GRT$, where

- T is a transformation that moves $\boldsymbol{p}_0$ to the origin,
- R is a rotation about the origin taking the epipole to a point on the u –axis, and
- $G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ {}^{-1}/_{f} & 0 & 1 \end{bmatrix}$

The matching transformation

- Let $\boldsymbol{p}$ and $\boldsymbol{p}'$ be a pair of corresponding points.
- Recall that $\mathcal{F}\boldsymbol{p}'$ represents the epipolar line on the left image induced by point $\boldsymbol{p}'$ on the right image.
- After the left image is ressampled by H, the new epipolar line will be $H^{-T}\mathcal{F}\boldsymbol{p}'$ (why?)
- The mapping matrix H' to be applied to the right image must be such that

horizontal epipolar line on the right image

$$H'^{-T}\mathcal{F}^T\boldsymbol{p} = H^{-T}\mathcal{F}\boldsymbol{p}'$$

horizontal epipolar line on the left image

- Many matching transformations satisfy this condition.

Algorithm outline

i. Identify a set of image-to-image matches
ii. Compute the fundamental matrix $\mathcal{F}$ and find the epipoles **e** and **e'**.
iii. Select the projective transformation H that maps the epipole **e** to the point at infinite $[\,f, 0, 0]^T$
iv. Find the matching projective transformation H' that minimizes the relative displacement of matching points along the u axis (distortion)

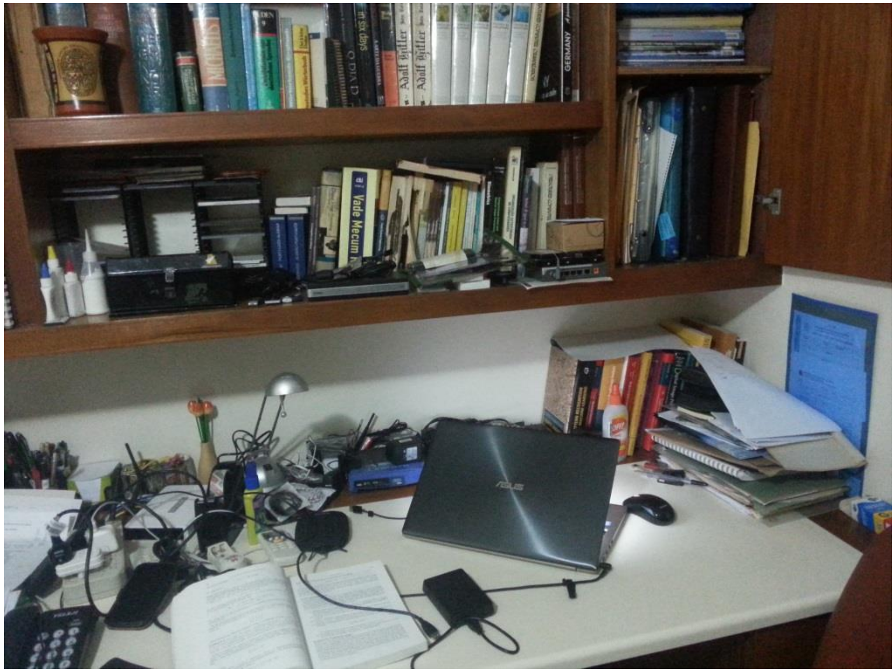
$$\sum_i d(H\boldsymbol{p}_i, H'\boldsymbol{p}'_i)$$

v. Resample both images according to H and H'.

Details can be found in
Multiple View Geometry in computer Vision, Hartley,R. and Zisserman, A., section 11.12 pp 302…

Example

Input Images



Example

Rectified Images

Rectification

Remarks:

- The search for corresponding points becomes 1D.
- It is possible to find a match for almost every pixel → dense match map.
- Compensates rotation over the image plane, which improves cross correlation.
- Reconstruction can be done from rectified images.

Assignments

RECONSTRUCTION:

Download the MATLAB (IteractiveTriangulation) program that demonstrates stereo reconstruction interactively. Experiment with different reconstruction approaches. Run the TextureOverlayDemo program and experiment with the 3D model.

REGION GROWING + NORMALIZED CROSS CORRELATION:

Download the MATLAB demo program for Region Growing. Choose a pair of stereo images and experiment with different parameter values. Important: follow the instructions in the read.me file before running the program.

RECTIFICATION:

Download the MATLAB demo program for uncalibrated image rectification. Just run the demo using a stereo pair as input.

Read the MATLAB documentation and follow the examples on **Uncalibrated Stereo Image Rectification.**

Next Topic

More on Calibration